AMENDMENTS TO THE CLAIMS

1. (Currently Amended) [[~~A~~]] An in-order multi-threading processor, comprising:

a first instruction fetch unit to receive a first thread and a second instruction fetch unit to receive a second thread;

an execution unit to execute said first thread and said second thread in parallel; and

a multi-thread scheduler coupled to said first instruction fetch unit, said second instruction fetch unit, and said execution unit, wherein said multi-thread scheduler is to determine the width of said execution unit.

2. (Currently Amended) [[~~A~~]] An in-order multi-threading processor as recited in claim 1, wherein [[~~the~~]] said multi-thread scheduler unit determines whether [[~~the~~]] said execution unit is to execute [[~~the~~]] said first thread and [[~~the~~]] said second thread in parallel depending on the width of [[~~the~~]] said execution unit.

3-4 (Cancelled)

5. (Currently Amended) [[~~A~~]] An in-order multi-threading processor as recited in claim [[~~3~~]] 2, wherein [[~~the~~]] said execution unit executes ~~the first~~ a third thread and ~~the second~~ a fourth thread in series.

6. (Currently Amended) [[~~A~~]] An in-order multi-threading processor as recited in claim [[~~3~~]] 2, wherein [[~~the~~]] said first thread and [[~~the~~]] said second thread are compiled to have instruction level parallelism.

7. (Currently Amended) [[~~A~~]] An in-order multi-threading processor as recited in claim 6, further comprising:

a first instruction decode unit coupled between [[~~the~~]] said first instruction fetch unit and [[~~the~~]] said multi-thread scheduler; and

a second instruction decode unit coupled between [[~~the~~]] said second instruction fetch unit and [[~~the~~]] said multi-thread scheduler.

8. (Currently Amended) [[~~A~~]] An in-order multi-threading processor as recited in claim 4, wherein [[~~the~~]] said execution unit executes only two threads in parallel.

9. (Currently Amended) A computer implemented method, comprising:

determining whether [[~~a~~]] an in-order multi-threading processor is wide enough to execute a first thread and a second thread in parallel; and

executing said first thread and said second thread in parallel if said in-order multi-threading processor is wide enough to execute [[~~the~~]] said first thread and [[~~the~~]] said second thread in parallel.

10. (Currently Amended) The method as recited in claim 9, further comprising executing [[~~the~~]] said first thread and [[~~the~~]] said second thread in series if said in-order multi-threading processor is not wide enough.

11. (Cancelled)

12. (Currently Amended) The method as recited in claim [[~~11~~]] 9, further comprising compiling the first thread and the second thread, wherein the first thread and the second thread have instruction level parallelism.

13. (Currently Amended) The method as recited in claim 12, wherein [[~~the~~]] said multi-threading processor executes only two threads in parallel.

14. (Currently Amended) The method as recited in claim 13, further comprising:

fetching [[~~the~~]] said first thread and [[~~the~~]] said second thread; and

decoding [[~~the~~]] said first thread and [[~~the~~]] said second thread.

15. (Currently Amended) A set of instructions residing in a storage medium, said set of instructions to be executed by [[~~a~~]] an in-order multi-threading processor for searching data stored in a mass storage device comprising:

determining whether said in-order multi-threading processor is wide enough to execute a first thread and a second thread in parallel; and executing said first thread and said second thread in parallel if said multi-threading processor is wide enough to execute [[~~the~~]] said first thread and [[~~the~~]] said second thread in parallel.

16. (Currently Amended) A set of instructions as recited in claim 15, further comprising executing [[~~the~~]] said first thread and [[~~the~~]] said second thread in series if said in-order multi-threading processor is not wide enough.

17. (Cancelled)

18. (Currently Amended) A set of instructions as recited in claim [[~~17~~]] 16, further comprising compiling [[~~the~~]] said first thread and [[~~the~~]] said second thread, wherein [[~~the~~]] said first thread and [[~~the~~]] said second thread have instruction level parallelism.

19. (Currently Amended) A set of instructions as recited in claim 18, wherein [[~~the~~]] said in-order multi-threading processor executes only two threads in parallel.

20. (Currently Amended) A set of instructions as recited in claim 19, further comprising:

fetching [[~~the~~]] said first thread and [[~~the~~]] said second thread; and

decoding [[~~the~~]] said first thread and [[~~the~~]] said second thread.

21. (Currently Amended) A system comprising:

a memory to store a set of instructions; and

[[~~a~~]] an in-order processor coupled to the memory to execute [[~~the~~]] said set of instructions, [[~~the~~]] said in-order processor with a first instruction fetch unit to receive a first thread, a second instruction fetch unit to receive a second thread, an execution unit to execute said first thread and said second thread, and a multi-thread scheduler coupled to said first instruction fetch unit, said second instruction fetch unit, and said execution unit, wherein said multi-thread scheduler is to determine the width of said execution unit.

22. (Currently Amended) The system of claim 21, wherein [[~~the~~]] said multi-thread scheduler unit determines whether [[~~the~~]] said execution unit is to execute [[~~the~~]] said first thread and [[~~the~~]] said second thread in parallel depending on the width of [[~~the~~]] said execution unit.

23. (Cancelled)

24. (Currently Amended) The system of claim [[~~23~~]] 22, wherein [[~~the~~]] said execution unit executes [[~~the~~]] said first thread and [[~~the~~]] said second thread in parallel.

25. (Currently Amended) The system of claim [[~~23~~]] 22, wherein [[~~the~~]] said execution unit executes [[~~the~~]] said first thread and [[~~the~~]] said second thread in series.

26. (Currently Amended) The system of claim [[~~23~~]] 22, wherein [[~~the~~]] said first thread and [[~~the~~]] said second thread are compiled to have instruction level parallelism.

27. (Currently Amended) The system of claim 26, further comprising:

a first instruction decode unit coupled between [[~~the~~]] said first instruction fetch unit and [[~~the~~]] said multi-thread scheduler; and

a second instruction decode unit coupled between [[~~the~~]] said second instruction fetch unit and [[~~the~~]] said multi-thread scheduler.

28. (Currently Amended) The system of claim 24, wherein [[~~the~~]] said execution unit executes only two threads in parallel.